PRIVATE AUDIO DELIVERY IN REVERBERANT SPACES

BY

YU-JEH LIU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Assistant Professor Ivan Dokmanić

ABSTRACT

We study the problem of delivering private audio to multiple users in a reverberant room. New methods are proposed which work with specially designed noise signals and make constructive use of reverberation.

First, traditional sound zones problem formulation is introduced. In this formulation, a loudspeaker array is used to create isolated soundfields as designated by the user. Traditional approaches such as acoustic contrast control (ACC) and pressure matching (PM) were devised as solutions to the sound zones problem, and they were proven to be effective. One aspect that has not been addressed by these methods is the privacy/security issue. With eavesdroppers present in the room, traditional approaches to sound zones problems do not ensure secure delivery of audio that prevents the eavesdroppers from comprehending the contents. Next, new methods are thus introduced which address the added privacy requirement.

Instead of considering reverberation as unwelcome, the proposed formulation takes advantage of the multi-path nature and leverages the random-like echoes to simultaneously achieve sound focusing and eavesdropping prevention. Two methods are introduced. The first one is based on direct least-squares optimization; we refer to it as the Least-Squares (LS) method. The second one, adopted from the wireless communication literature, exploits the null space of the multiple-input-multiple-output channel matrix; we refer to it as the Null Space (NS) method.

The two methods are evaluated and compared in numerical and real-world experiments. Both methods are shown to achieve sound focusing as well as very low signal-to-noise ratio (SNR) outside the focusing spots. The NS method provides sharper intelligibility drop which results in smaller, more spatially refined sound fo-

cusing spots.

Subject to the usual limitations of sound zone methods such as long computation time or the requirement to know the impulse responses, both methods are proven to provide new solutions to sound zone problems with privacy constraints.

To my family and my dear friends, for their love and support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Private Audio

The idea of private audio refers to reproducing different sounds in different spatial zones such that the interference is minimized among the zones, and therefore, different users in these corresponding zones are only able to hear the designated sound. The spatial zones of interest are referred to as the sound zones. Figure 1.1 shows a common situation that takes place in a living room. In this setting, in order to minimize the interference, achieving private audio will be essential for Joe (in the yellow oval) who is listening to music and Jack and Jane (in the green oval) who are watching the newest episode of their favorite TV show.

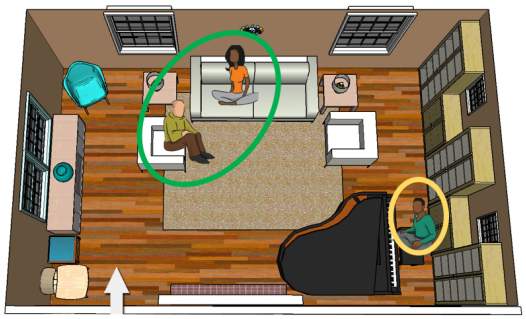

Figure 1.1: Illustration for a typical situation in which private audio is helpful

In many previous research efforts, an array of loudspeakers is used to deliver audio signals to corresponding spatial zones while attempting to achieve silence

outside these zones of interest. In general, sound zone problems can be thought of having two main components. One is the problem of the creation of the isolated zones, and the other is the problem of accurate acoustic reproduction. Two main approaches are developed to address, to different degrees, the two aspects of sound zone components just mentioned. These are the acoustic contrast control (ACC) method and the pressure matching (PM) method [1, 2, 3, 4].

For sound zone problems, it is typical to assume that there are only two spatial zones of interest. One of them will receive a sound signal, and the other will have silence. These two zones are referred to as the bright zone and the dark zone. The two-zone formulation can be easily adapted to having multiple zones.

1.2 Acoustic Contrast Control

In ACC method formulation, the goal is to maximize the sound pressure level contrast between the bright zone and the dark zone using L loudspeakers. In other words, we can write it as a maximization problem as follows

$$\max_{g} \|H_b g\|_2^2 \quad \textit{such that } \|H_d g\|^2 \leq D_0,\ \|g\|^2 \leq E_0,$$

where $g(\omega)$ is the loudspeakers' weights vector with size $L \times 1$. It depends on the driving signal's frequency ω. H_b and H_d are transfer functions describing the sound propagation from the loudspeakers to the bright zone and the dark zone, respectively. D_0 and E_0 are the upper limits we set for the sound pressure level at the dark zone and for the array power consumption.

When compared with a traditional sound focusing method, such as the time-reversal method [5] or the delay-sum method, the ACC method is proven to provide superior sound pressure level contrast [2]. Because the ACC formulation only maximizes the sound pressure level contrast but does not optimize the acoustic reproduction accuracy, it requires low array power consumption [6] and is useful for a smaller system (with few loudspeakers) with which precise reproduction of a soundfield is

not required.

1.3 Pressure Matching

In contrast to the ACC method which only aims to maximize the sound level contrast between the zones, in order to reproduce a perceptually accurate soundfield, the PM formulation is needed. For the PM method, the objective function is shown in the following equation. In this case, the function is minimized instead of maximized:

$$\min_{g} \|H_b g - p_{des}\|_2^2 \quad \textit{such that } \|H_d g\|^2 \leq D_0,\ \|g\|^2 \leq E_0,$$

where in addition to the terms introduced in the ACC method, p_{des} is the desired sound pressure level at the bright zone.

Instead of maximizing the sound pressure level contrast between the two zones like the ACC method, the PM method strives to minimize the difference between the generated soundfield and a desired soundfield in the bright zone. The overall soundfield in the bright zone is modeled as a superposition of soundfields induced by separate propagating planewaves [1, 7, 8]. While early studies focus on free-field propagation [9, 4, 6], in recent years a number of techniques were developed that address the effects of room acoustics such as reverberation, scattering, and absorption [10, 11].

Matching the desired soundfield in the bright zone is power-hungry: the array power consumption of the PM method is several times greater than for the ACC method [6]. This fact implies that with the same settings, i.e., same number of zones and same sizes of zones, a larger system with more loudspeakers is required for the PM formulation.

1.4 Limitations and Drawbacks of Traditional Sound Zone Problems

Although the above prior works on ACC and PM methods are tested and proven to be capable of generating desired sound zones, they have certain drawbacks which we list below.

First, depending on the direction of arrival of the propagating soundwaves and the locations of the bright and dark zones, an occlusion might happen and decrease the performance of the methods considerably [1]. Occlusion refers to the situation when the two sound zones are aligned with the direction of a propagating wave; one zone will then "occlude" the other. In such a situation, either silence in the dark zone or the desired soundfield in the bright zone would be difficult to achieve.

Second is the issue of spatial aliasing. When an array of loudspeakers is used to generate a sound wave, spatial aliasing happens when the nearest two loudspeakers are placed more than one half of the wavelength apart. It causes angular ambiguity and thus large errors in the generated soundfield. Distances between loudspeakers thus need to be carefully adjusted to avoid spatial aliasing.

Third, a large number of loudspeakers are usually required. Using the PM method, the number of loudspeakers required grows quadratically with the frequency of the reproduced soundwave [9]. For example, it requires around 100 loudspeakers to create a 3,000 Hz plane wave inside the bright zone with low errors and high contrast [6].

Fourth is the inflexibility of the loudspeaker array. In prior studies, the shape of the loudspeaker array is fixed as spherical, circular or linear due to the parametric modeling of the desired soundfield. For an array of arbitrarily placed loudspeakers, there is no general solution.

Fifth, free-field propagation is the assumed situation for most of the methods. Echoes created by the walls are considered as deteriorating effects that worsen the

performance of the methods. Specifically, the soundfield generated by the reverberant environment needs to be approximated using a specific and delicate mathematical model, and this approximation may be fragile and inaccurate enough to introduce large errors when producing the sound zones [10, 11].

Lastly and most importantly for our work, the unattended regions, area excluding the dark and bright zones, are generally not of interest to the methods. As a consequence, listeners in these regions will most likely be able to comprehend the sound signals well. Because the loudspeaker signals are generated by linear time-invariant filtering of the target waveforms, those signals will resemble the target waveforms and thus be comprehensible everywhere.

1.5 Motivations

Different situations require different solutions. The limits and drawbacks of sound zones should motivate us to consider the situation when unwanted listeners are present in the unattended area in a confined space. To be specific, when speech signals are to be privately delivered to different people in a room where eavesdroppers are also present, the resulting audio in the unattended area must be taken into consideration. The goal of the methods proposed in this thesis is to have only the listeners in one specific zone understand their designated speech.

As long as the sound signals delivered are unintelligible outside the designated zones, the goal is achieved. Therefore, explicitly constructing silence zones, as in the sound zone problems, is not needed nor desired anymore. As a result, instead of superposing soundwaves to achieve isolated sound zones, jointly optimizing the loudspeakers' driving signals such that various sound signals can be delivered only to their respective zones becomes a reasonable new formulation.

Continuing with this thought, in this thesis we explore and compare methods to create acoustic zones for private communications. In each zone, delivery of one distinct speech signal is expected, and the listeners outside the particular zone

are not expected to understand the message. We propose two distinct methods. The first one achieves privacy by introducing non-linear segmentation of the speech signals and solving a least-squares error minimization problem. The second proposed method is a nullspace cancellation method which works by emitting specially designed noise.

CHAPTER 2

NEW FORMULATIONS FOR PRIVATE AUDIO MESSAGE

In this chapter, different approaches are proposed to solve the sound zone problems. In the first part of this chapter, a new mathematical formulation of the sound zone problem is derived. The Least-Squares (LS) method is introduced, and the solution is obtained. Results from both simulation and real-world experiments are shown. In the second part of this chapter, the Null Space (NS) method is introduced and compared with the LS method in terms of the performance demonstrated for both simulation and real-world experiments. At the end of this chapter, future work is mentioned.

2.1 Problem Formulation

Basic Model for K Listeners

Suppose K private messages are to be received by K listeners in the same confined space using L loudspeakers, such as the case shown in Fig. 2.1 for $K = 2$. Denoting the signal emitted by the ℓth loudspeaker by $s_\ell[n]$, the received audio signal $y_k[n]$ for the kth user can be formulated as

$$y_k[n] = \sum_{\ell=1}^{L} s_\ell[n] * h_{k\ell}[n], \tag{2.1}$$

where the $*$ operator represents a linear convolution and $h_{k\ell}[n]$ is the overall impulse response from the ℓth loudspeaker to user k.

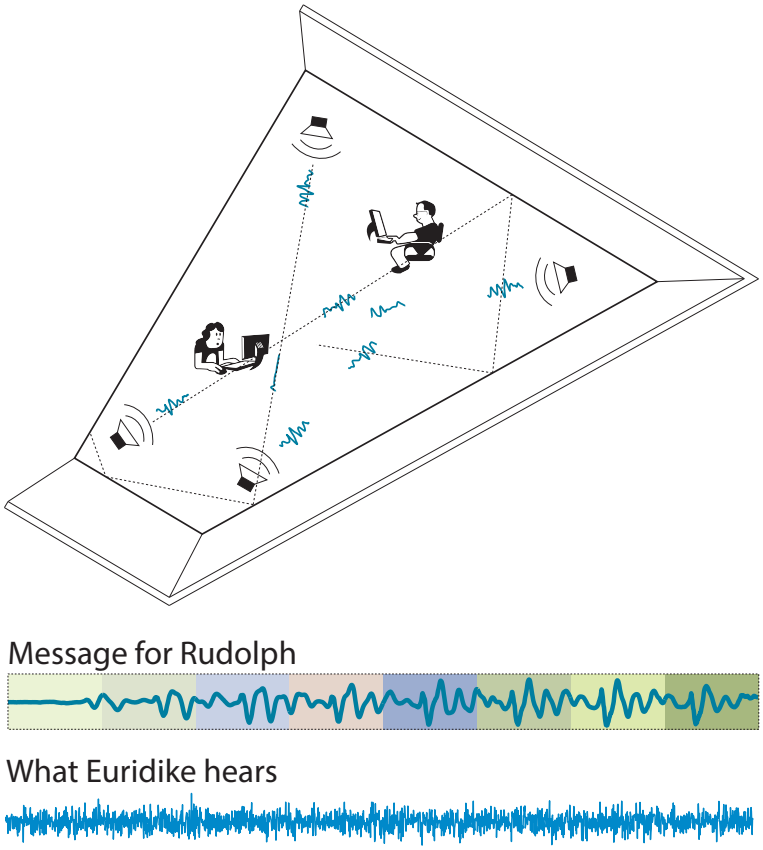

Figure 2.1: An illustration of the sound zone problem

Notice here the overall impulse response $h_{k\ell}[n]$ can be thought of as the convolution between the electro-acoustic response of the ℓth loudspeaker $hs_\ell[n]$ and the room impulse response (RIR) $hr_{k\ell}[n]$ from the ℓth loudspeaker to location k. Therefore,

$$h_{k\ell}[n] = hs_\ell[n] * hr_{k\ell}[n]. \tag{2.2}$$

Naive Inverse Filter Method

Due to reverberation from the walls, the received signals $y_k[n]$ are linearly distorted. To recover the original signal, one method is to pre-filter the original signal with a designed filter that is essentially the inverse filter of the overall impulse response $h_{k\ell}[n]$. In that case, the ℓth speaker's pre-filtered driving signal can be written as

$$s_\ell[n] = x_k[n] * f_{k\ell}[n], \tag{2.3}$$

where $x_k[n]$ is the original audio signal intended to be received by the user at location k and $f_{k\ell}[n]$ is the filter designed to be the inverse of the overall impulse response $h_{k\ell}[n]$.

With a substitution of Eq. (2.3) into Eq. (2.1), the received signal can be written as

$$y_k[n] = \sum_{\ell=1}^{L} (x_k[n] * f_{k\ell}[n] * h_{k\ell}[n]) \tag{2.4}$$

such that it satisfies the condition

$$f_{k\ell}[n] * h_{k\ell}[n] = \delta[n - T], \tag{2.5}$$

where T represents some amount of delay, or equivalently, in the frequency domain,

$$F_{k\ell}[\omega] * H_{k\ell}[\omega] = 1 \cdot e^{-j\omega T}. \tag{2.6}$$

The above conditions are true since the received signal $y_k[n]$ is expected to be a delayed version of the original signal $x_k[n]$.

However, this operation can lead to an unstable system depending on the overall impulse response $h_{k\ell}[n]$. For one example, if $h_{k\ell}[n]$ had nulls (or values close to null) in the frequency domain at some specific values, the inverse filter $f_{k\ell}[n]$ would not be bounded. Thus, the overall system response becomes unstable and far from $\delta[n-T]$ whenever even small errors are present.

Therefore, instead of explicitly designing inverse filters for each individual channel, a multi-channel approach is proposed in the next section to achieve the preservation of the original signal $x_k[n]$ using the notion of least-squares error minimization.

Segments of Signals: Temporal Chopping

After least-squares error minimization is considered for accurate reproduction of the sound signal, the other essential part of the sound zone problem, the creation of isolated zones, must be addressed as well.

Most of the multi-channel approaches feed in the same sound signal as the original input for all loudspeakers, and different weight functions (filters) are applied to the input to obtain different driving signals for different loudspeakers [1, 2, 3, 8, 12]. The approaches are proven to have excellent sound focusing ability. However, because the loudspeakers are driven by filtered versions of the target signal, listeners in the unattended zone are able to comprehend the content of the sound signals as well. Intuitively, to exploit the advantage of having multiple channels, we propose the idea of using discontinuous sound signals as the input sound signals in hope of further disrupting the response in the unattended area. This can be achieved using so-called multiplicative masks $w_{k\ell}$ on $x_k[n]$.

For every $k \in \{1, \ldots, K\}$, we produce L masks $\{w_{k\ell}\}_{\ell=1}^{L}$ and assign masked signals $\tilde{x}_{k\ell}[n] = x_k[n]w_{k\ell}[n]$ to the ℓth loudspeaker after adequate linear, shift-invariant (LSI) filtering. We refer to $\tilde{x}_{k\ell}$ as the *design signal*.

Multiplicative masks are designed to segment every user message into L

submessages assigned to each of the L loudspeakers. This can be thought of as scrambling pieces of the sound signals around the confined space and putting them back together at specific locations via precise computation. The idea is illustrated in Fig. 2.2.

However, if the chopped segments are too short or the window used to divide the signal into segments is discontinuous, the recombined message will contain unpleasant audible artifacts. This is partly due to the non-ideal electroacoustical response of the loudspeakers.

A better idea is to segment the signals by smooth, overlapping windows that rise and fall over T samples, and flatten out over D samples. An example of such a smooth mask is illustrated in Fig. 2.3. We refer this application of smooth mask as temporal chopping.

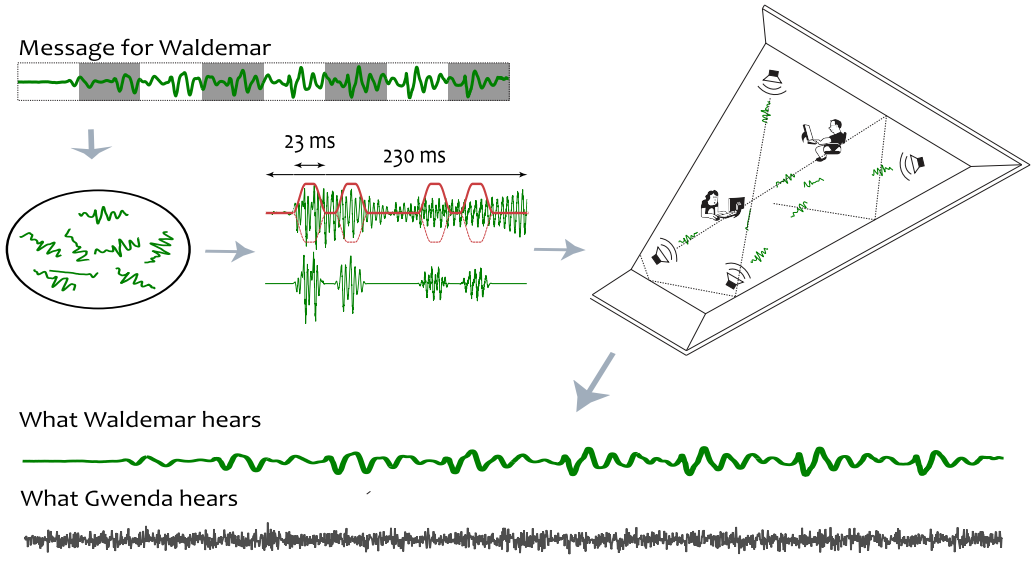

Figure 2.2: Basic idea of taking the sound signals apart first and recombining them at locations of interest

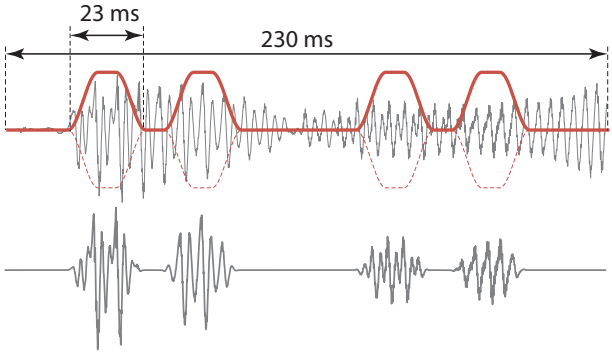

Figure 2.3: Example for temporal chopping done on a pure sine signal

Namely, each of the multiplicative masks has a cosine profile such that

$$w_{k\ell}[n] = \begin{cases} \cos^2\left[\frac{\pi}{2}\left(1-\frac{n}{T-1}\right)\right], & 0 \le n < T \\ 1, & T \le n < T+D \\ \cos^2\left[\frac{\pi}{2}\left(1-\frac{n-(T+D)}{T-1}\right)\right], & T+D \le n < 2T+D \end{cases} \tag{2.7}$$

which is a variation on the Tukey window.

L such smooth masks are generated for each of the signal k ensuring that they sum to a constant,

$$\sum_{\ell=1}^{L} w_{k\ell}[n] = 1, \quad \forall k \in \{1,\ldots,K\}, n \in \{0,\ldots,N-1\}. \tag{2.8}$$

The logic behind Eq. (2.8) is that in the anechoic case, simply reproducing adequately delayed and amplified signals would achieve the desired effect since it ensures that the sum of the design signals equals the original signal

$$\sum_{\ell=1}^{L} \tilde{x}_{k\ell}[n] = x_k[n]. \tag{2.9}$$

With the temproal chopping, instead of directly reproducing $\tilde{x}_{k\ell}[n]$ with the ℓth loudspeaker, we first filter it with an LSI filter $g_{k\ell}[n]$. The role of $g_{k\ell}[n]$ is to adjust the phases of the emitted chunks of the design signal $\tilde{x}_{k\ell}[n]$ so that after reverberations inside the room, the segments align at specific loctaions such as where Waldemar is in Fig. 2.2. Designing these filters $g_{k\ell}[n]$ is the main computational step in our proposed method.

In later sections, results are provided to showcase the effectiveness of the temporal chopping on distorting the received signal in the unattended area. For example, in the case described by Fig. 2.2, Gwenda hears only noise at her location.

Final LS Model with Temporal Chopping

Now with the entire setup in place, the proposed mathemetical formulation can be presented. To start, the LS problem to solve can be written as

$$\min_{y_k[n]} \sum_{k=1}^{K} \|\xi_k[n] - y_k[n]\|^2, \tag{2.10}$$

where $\xi_k[n]$ is simply a delayed version of the original sound signals $x_k[n]$ to be delivered.

Second, when the temporal chopping and the LSI filter $g_{k\ell}[n]$ are applied, the emitted sound signal $s_\ell[n]$ becomes

$$s_\ell[n] = \sum_{k=1}^{K} \tilde{x}_{k\ell}[n] * g_{k\ell}[n] = \sum_{k=1}^{K} \big\{x_k[n] w_{k\ell}[n]\big\} * g[n]. \tag{2.11}$$

Combined with Eq. (2.1), the final received signals $y_k[n]$ have the form

$$y_k[n] = \sum_{\ell=1}^{L} s_\ell[n] * h_{k\ell}[n] = \sum_{\ell=1}^{L} \sum_{k'=1}^{K} \tilde{x}_{k'\ell}[n] * g_{k'\ell}[n] * h_{k\ell}[n]. \tag{2.12}$$

Futhermore, we look at finite-length signals of N samples and define the following vectors

$$\begin{aligned}
\mathbf{x}_k &\stackrel{\text{def}}{=} \begin{bmatrix} x_k[0], & x_k[1], & \ldots, & x_k[N-1] \end{bmatrix}^\top \\
\mathbf{w}_{k\ell} &\stackrel{\text{def}}{=} \begin{bmatrix} w_{k\ell}[0], & w_{k\ell}[1], & \ldots, & w_{k\ell}[N-1] \end{bmatrix}^\top \\
\mathbf{g}_{k\ell} &\stackrel{\text{def}}{=} \begin{bmatrix} g_{k\ell}[0], & g_{k\ell}[1], & \ldots, & g_{k\ell}[M-1] \end{bmatrix}^\top \\
\mathbf{h}_{k\ell} &\stackrel{\text{def}}{=} \begin{bmatrix} h_{k\ell}[0], & h_{k\ell}[1], & \ldots, & h_{k\ell}[P-1] \end{bmatrix}^\top.
\end{aligned} \tag{2.13}$$

Since convolution in discrete time can be expressed as a matrix multiplication with a Toeplitz matrix, we can construct a matrix which is a $(N+M-1)\times M$ Toeplitz matrix that corresponds to a linear convolution of $\mathbf{x}_k$ and a signal of length M

$$\tilde{\mathbf{X}}_{k\ell} = \begin{pmatrix} \tilde{x}_{k\ell}[0] & 0 & 0 & 0 & \cdots & 0 \\ \tilde{x}_{k\ell}[1] & \tilde{x}_{k\ell}[0] & 0 & 0 & \cdots & 0 \\ \tilde{x}_{k\ell}[2] & \tilde{x}_{k\ell}[1] & \tilde{x}_{k\ell}[0] & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \tilde{x}_{k\ell}[N-1] & \tilde{x}_{k\ell}[N-2] \\ 0 & 0 & 0 & \cdots & 0 & \tilde{x}_{k\ell}[N-1] \end{pmatrix}. \tag{2.14}$$

For concise math expression, we will write $\tilde{\mathbf{X}}_{k\ell} = \mathsf{Toeplitz}\big(\tilde{x}_{k\ell}[n]\big) = \mathsf{Toeplitz}\big(\mathsf{diag}(\mathbf{w}_{k\ell})\mathbf{x}_k\big)$ for short. The signal driving the ℓth loudspeaker can then be written as

$$\mathbf{s}_\ell = \sum_{k=1}^{K} \tilde{\mathbf{X}}_{k\ell}\mathbf{g}_{k\ell} = \tilde{\mathbf{X}}_\ell \mathbf{g}_\ell, \tag{2.15}$$

where $\tilde{\mathbf{X}}_\ell = \big[\,\tilde{\mathbf{X}}_{1\ell}, \tilde{\mathbf{X}}_{2\ell}, \ \ldots, \tilde{\mathbf{X}}_{K\ell}\,\big]$, $\mathbf{g}_\ell = \big[\,\mathbf{g}_{1\ell}^\top, \ \mathbf{g}_{2\ell}^\top, \ \ldots, \mathbf{g}_{K\ell}^\top\,\big]^\top$.

Finally, combining the convolution with the overall impulse response $h_{k\ell}[n]$, the kth user receives the signal $y_k[n]$ in vector form as

$$\mathbf{y}_k = \sum_{\ell=1}^{L} \mathbf{H}_{k\ell}\mathbf{s}_\ell = \mathbf{H}_k\tilde{\mathbf{X}}\mathbf{g}, \tag{2.16}$$

with

$$\begin{cases} \mathbf{H}_{k\ell} = \mathsf{Toeplitz}(\mathbf{h}_{k\ell}) \\ \tilde{\mathbf{X}} = \mathsf{blockdiag}\big(\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_L\big) \\ \mathbf{g} = \big[\, \mathbf{g}_1^\top,\ \mathbf{g}_2^\top,\ \ldots, \mathbf{g}_L^\top \,\big]^\top. \end{cases} \tag{2.17}$$

For all K users in a single matrix-vector equation, we get

$$\mathbf{y} = \mathbf{H}\tilde{\mathbf{X}}\mathbf{g}, \tag{2.18}$$

with $\mathbf{y} = \big[\, \mathbf{y}_1^\top,\ \mathbf{y}_2^\top,\ \ldots, \mathbf{y}_K^\top \,\big]^\top$, $\mathbf{H} = \big[\, \mathbf{H}_1^\top,\ \mathbf{H}_2^\top,\ \ldots, \mathbf{H}_K^\top \,\big]^\top$.

The task now becomes finding the very long filter vector $\mathbf{g} \in \mathbb{R}^{(MLK)\times 1}$. With the LS problem stated by Eq. (2.10), the private sound delivery problem becomes a minimization problem as

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}} \|\xi - \mathbf{H}\tilde{\mathbf{X}}\mathbf{g}\|^2. \tag{2.19}$$

One can observe that it is essentially equal to solving a system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} = \mathbf{H}\tilde{\mathbf{X}}$, $\boldsymbol{x} = \mathbf{g}$ and $\boldsymbol{b} = \xi$.

As explained in Appendix A, the solution is in principle given as $\hat{\mathbf{g}} = (\mathbf{H}\tilde{\mathbf{X}})^\dagger \xi$, where $(\cdot)^\dagger$ is the Moore-Penrose pseudoinverse. However, because of the high sampling rate of the audio signals which contributes to large N, the involved matrices are far too large for direct computation of the pseudoinverse. This issue is what drives us towards the iterative methods mentioned in the previous section. In this case, in order to achieve faster convergence rate, we use the conjugate gradient method. Since both $\mathbf{H}\tilde{\mathbf{X}}$ and the adjoint $\tilde{\mathbf{X}}^\top\mathbf{H}^\top$ consist of multiplications by convolution matrices, the conjugate gradient method can be efficiently implemented using fast Fourier transforms.

One thing to notice is that only the matrix $\mathbf{H}$ varies depending on the user's location k and the overall impulse responses from the ℓth loudspeakers to the location k. The matrix $\tilde{\mathbf{X}}$ is derived from all original signals $x_k[n]$ and the corresponding

temporal chopping window functions $w_{k\ell}[n]$. This separates the individual effects of the room geometry and of the design signals.

2.2 Initial Simulation Results

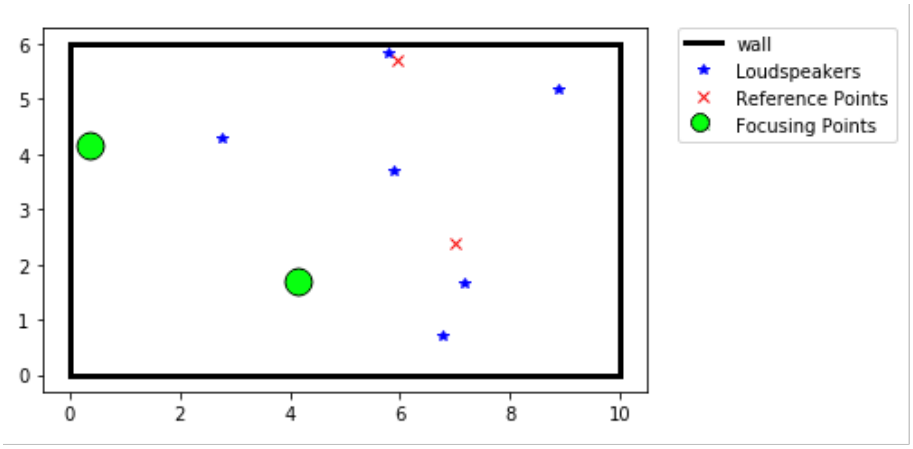

Figure 2.4: Spatial settings for the initial simulations

We now evaluate the performance of the above formulation through numerical simulations. The STOI metric [13] is used as the intelligibility score metric for measuring the significance of our results.

For the simulations, we use the Python room simulation package Pyroomacoustics [14]. The parameters of the simulation are set as follows:

- Six loudspeakers are used in a 10m $\times$ 6m rectangular room.
- Four microphones are used, two for the two sound zones each and the other two placed randomly in the room as references.
- Two speech signals, each four seconds long, were used as the target signals delivered at the two sound zones.

The actual spatial setting for the initial simulations is shown in Fig. 2.4.

In addition, following is a list of settings for the signals and for the number of iterations of the conjugate gradient method:

- Sampling rate for the signals and the system is set to be 44.1 kHz.
- Temporal chopping is set to be done randomly in between loudspeakers with one chunk of design signal being 1000 samples long.
- The filter length is set to be 100,000 samples long.
- The conjugate gradient descent method is run for 100 iterations to obtain the filter $\hat{\mathbf{g}}$.

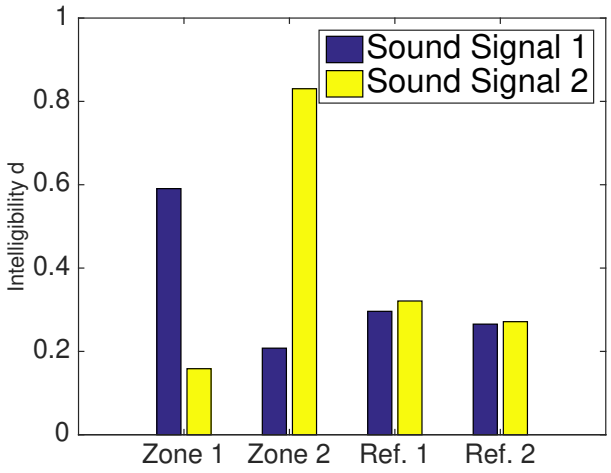

Figure 2.5: STOI intelligibility scores for the initial simulation

As it can be seen in Fig. 2.5, the difference of the STOI intelligibility scores between the two speech signals are substantial in both zones as intended. Also, the STOI intelligibility scores at the two randomly chosen reference points are low for both speech signals.

It is mentioned in the previous section that temporal chopping encourages sound distortion in the unattended area and thus can lower the intelligibility. We now verify through simulation that this is indeed the case. In this simulation, two

pure sine signals of different frequencies are the target signals to be delivered in the two zones. The received signals at the four locations are shown in Fig. 2.6 and Fig. 2.7.

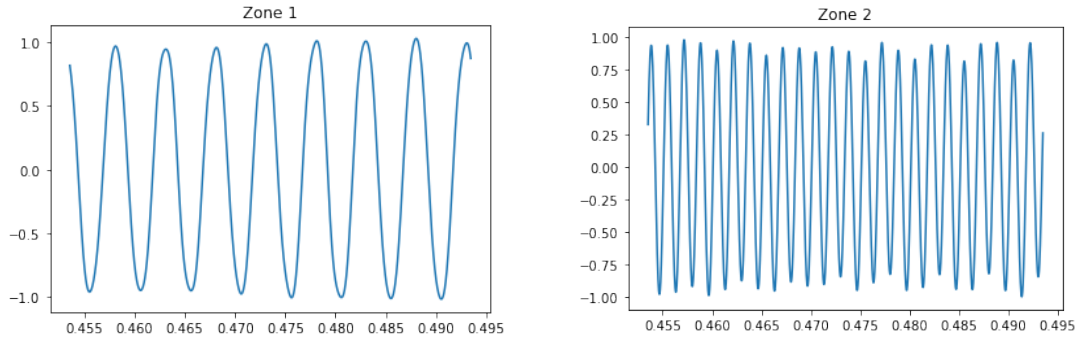

Figure 2.6: Received signals at the focusing zones

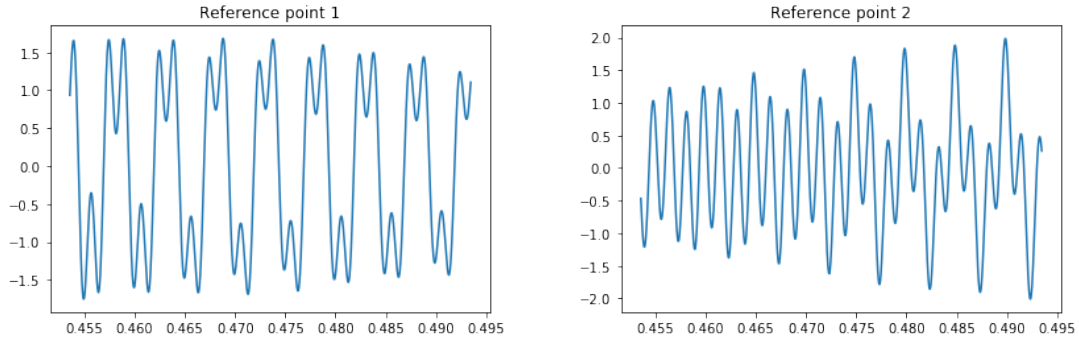

Figure 2.7: Received signals at the reference points in the unattended area

As we can see from the results, the designed filter $\hat{\mathbf{g}}$ does a good job aligning the temporal chopped inputs such that the received signals at the two zones are the two pure sine signals with small distortion. On the other hand, the received signals at the reference points show that the signals are distorted.

2.3 Improve with Further Randomization

The overall formulation and algorithm work well for creating isolated sound zones. We now explore possible improvements in the unattended area. From the above simulation experiments, it can be seen that the random temporal chopping indeed has the effect of lowering the intelligibility of the original message signals in the unattended area. In addition to that, in order to lower the intelligibility further in the unattended area, we considered other types of randomization.

Among the several kinds of randomizations attempted, we observed that one of them worked particularly well, both for raising the reproduction quality inside the zones of interest and for lowering the intelligibility in the unattended area. We refer to it as the Random Short-Time A-matrix Transform (Random STAT).

This randomization happens before the temporal chopping, and it is in the form

$$Z_{\ell k}[n] = [A^*(A\mathbf{x}_k \odot \mathbf{b}_\ell)][n], \tag{2.20}$$

where A is a random orthonormal matrix, A^* is the adjoint of A, x_k is the input signal for zone k, and b_ℓ is a binary mask for the ℓth speaker. The symbol $\odot$ denotes elementwise multiplication. Note that the binary masks will sum up to an all-ones vector. After the temporal chopping, A^* is applied to get the input signal for speaker ℓ to zone k,

$$s_\ell[n] = \sum_{k=1}^{K} (\mathbf{Z}_{\ell k} \odot \mathbf{w}_{\ell k})[n]. \tag{2.21}$$

We evaluate the above strategy via an FFT analysis by comparing performance of temporal chopping with and without Random STAT. The results are shown in Fig. 2.8 and Fig. 2.9.

From the figures, we observe that the Random STAT adds what resembles white Gaussian noise into our input signals, which evidently improves the overall performance of the proposed private audio delivery algorithm.

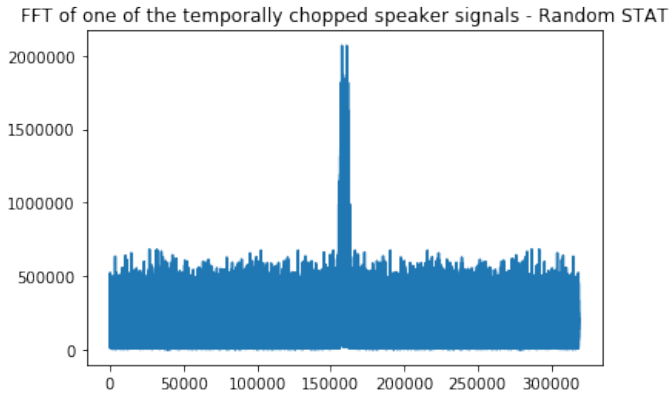

Figure 2.8: FFT spectra for one of the temporally chopped input with Random STAT

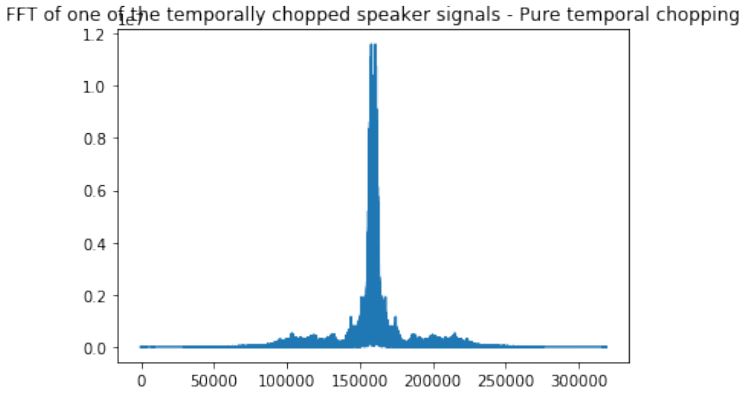

Figure 2.9: FFT spectra for one of the temporally chopped input without Random STAT

2.4 White Gaussian Noise as Inputs

Experiments in the last subsection suggest that adding white noise to the input signals improves the overall performance of the proposed algorithm. Intuitively, white noise adds higher frequency components that are outside the frequency range of typical speech. With more balanced spectra of the input signals, there will be more degrees of freedom to the system to produce a filter that approximates the

target signals at the focusing zones. In the following, we look at two figures of merit which corroborate the above discussion.

Coherence of the System Matrix

When the computed filters $g_{k\ell}[n]$ are poorly conditioned, any model mismatch such as tiny changes in room impulse responses will result in large errors in the signals received by the users. We can expect to get unsatisfactory filter responses when the system matrix $\mathbf{H}\tilde{\mathbf{X}}$ is poorly conditioned.

To quantify the conditioning of the system matrix $\mathbf{H}\tilde{\mathbf{X}}$, we measure the frequency-dependent coherence between randomly chosen pairs of columns in $\mathbf{H}\tilde{\mathbf{X}}$. For two signals $z[n]$, $w[n]$, coherence is defined as

$$\gamma_{zw}(f) = \frac{|C_{zw}(f)|^2}{A_{zz}(f)A_{ww}(f)}, \tag{2.22}$$

with C_{zw} being the Fourier transform of the crosscorrelation of z and w, and A_{zz}, A_{ww} Fourier transforms of their autocorrelations.

$\mathbf{H}\tilde{\mathbf{X}}$ has L blocks of columns, each corresponding to one loudspeaker. Columns in the same block are correlated as they are influenced by the impulse responses and driving signals, but it is desirable that the columns in different blocks be incoherent.

In Fig. 2.10a, the coherence between columns from *different* blocks is plotted. It is clear that using chopped speech without white noise as the design signal $\tilde{x}_{k\ell}$ gives a coherent $\mathbf{H}\tilde{\mathbf{X}}$ at many frequencies, while using chopped white noise gives low coherence.

On the other hand, we observe empirically that as soon as $\mathbf{H}\tilde{\mathbf{X}}$ has at least as many columns as rows, its row rank is full and the system has at least one solution.

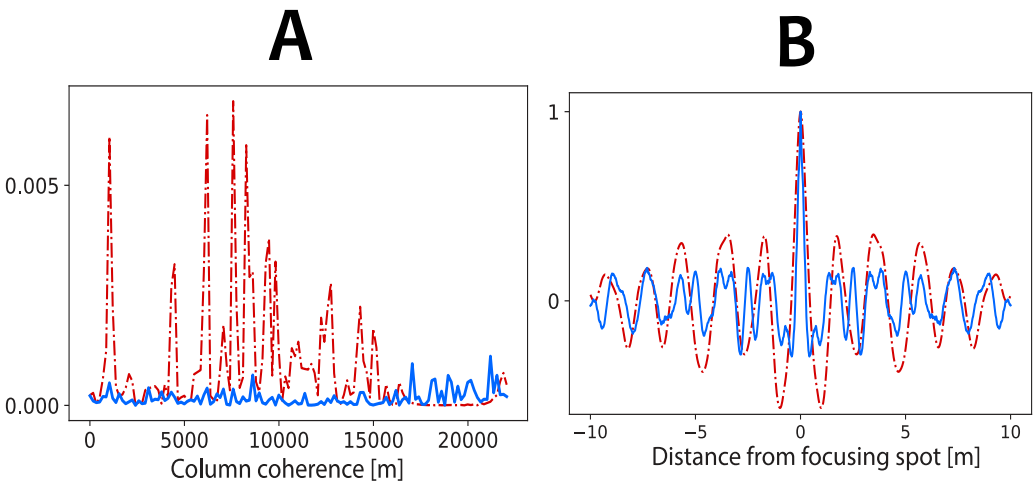

Figure 2.10: a) Coherence of two random columns from different blocks; b) autocorrelation with respect to distance; red is chopped speech with no white noise, and blue is chopped white noise [15]

This happens when

$$\underbrace{K(P+N+M-2)}_{\text{length of } \mathbf{y}} \leq \underbrace{KML}_{\text{length of } \mathbf{g}} \implies M(L-1) \geq P+N-2. \tag{2.23}$$

That is, the system has at least one solution as soon as the filters are long enough and that sufficiently many loudspeakers are present. This has a nice interpretation: in principle, we can convert chopped noise into any target signal via multichannel filtering. With chopped speech the matrix is close to being singular and the statement does not hold robustly.

The overall workflow of the proposed LS method for either chopped speech signals or chopped white noise is illustrated in Fig. 2.11.

Decay of the Autocorrelation

Another interesting observation is the faster decay of the autocorrelation of loudspeaker driving signals. This measures how fast the sound will decorrelate and

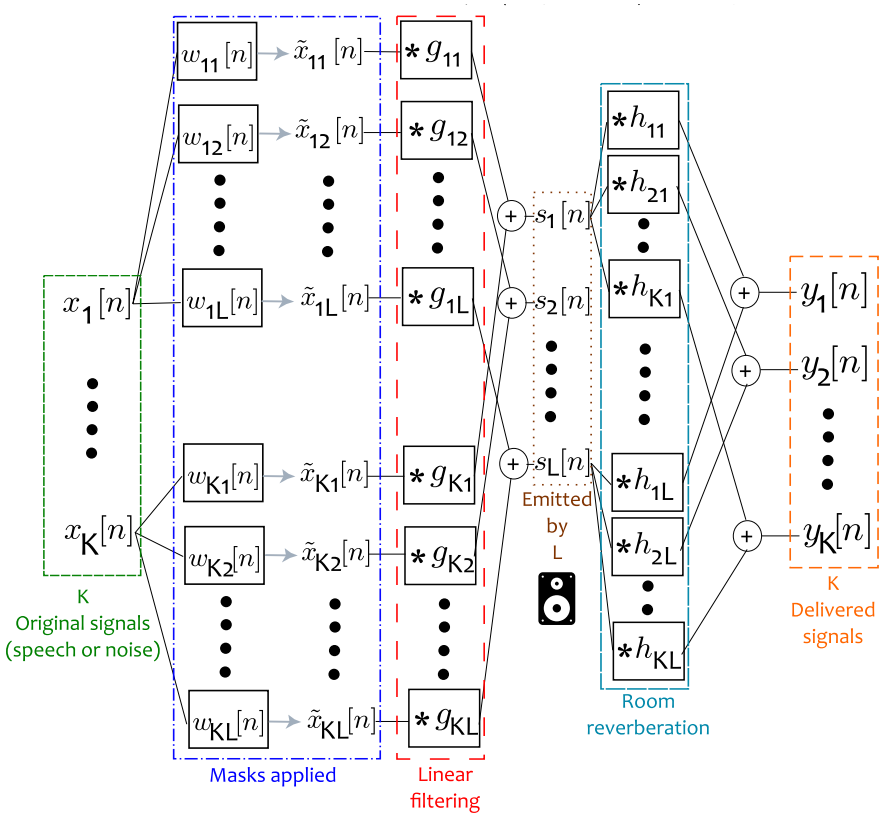

Figure 2.11: Workflow for the proposed LS method

become unintelligible as we move away from the focusing points. Figure 2.10b shows that using chopped white noise as the design signal $\tilde{x}_{k\ell}$ yields a faster decay of the autocorrelation than using chopped speech without white noise. To understand why, note that the autocorrelation $a_{s_\ell s_\ell}[n]$ of the emitted signal $s_\ell[n]$ can be written as

$$\begin{aligned} s_\ell[n]*s_\ell[-n] &= \sum_{k,k'} \tilde{x}_{k\ell}[n] * g_{k\ell}[n] * \tilde{x}_{k'\ell}[-n] * g_{k'\ell}[-n] \\ &= \sum_{k} a_{\tilde{x}_{k\ell}\tilde{x}_{k\ell}}[n] * a_{g_{k\ell}g_{k\ell}}[n] + \sum_{k\neq k'} c_{\tilde{x}_{k\ell}\tilde{x}_{k'\ell}}[n] * c_{g_{k\ell}g_{k'\ell}}[n]. \end{aligned} \tag{2.24}$$

The crosscorrelation $c_{\tilde{x}_k\tilde{x}_{k'}}[n]$ will depend on the signals used to feed the loudspeakers. In particular, we can expect that if we use noise, these crosscorrelations will

be small, and thus it reduces the overall autocorrelation of the loudspeaker driving signals.

The fact that using temporally chopped Gaussian white noise as inputs gives superior performance is further verified when the converging speed of the residual term is taken into consideration. With a fixed number of conjugate gradient iterations, results of varying quality at the focusing spots are obtained for the two cases. Figure 2.12 shows the value of the 2-norm square loss. At any given iteration, the approximation by using chopped speech as design signal is much worse than when we use chopped white noise. This is of course directly related to the condition number κ of the system matrix $\mathbf{H}\tilde{\mathbf{X}}$. It is well known that the number of iterations of conjugate gradient descent for a prescribed precision is proportional to $\sqrt{\kappa}$.

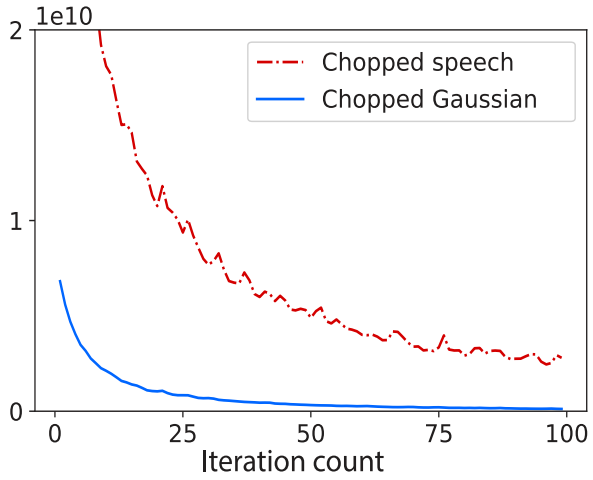

Figure 2.12: Residual for each conjugate gradient iteration; red is chopped speech with no white noise, and blue is chopped white noise [15]

2.5 Simulations for the White Noise Design

For the final simulation results, we again use Pyroomacoustics [14] along with the same figures and settings in the Initial Simulation Results subsection except that

we add in one extra reference point and target signals are seven seconds long.

As we can see in Fig. 2.13, the intelligibility scores for the sound signals delivered at the zones improve greatly along with the intelligibility difference between the two sound signals in a single zone. In addition, the intelligibility scores of the two signals in the unattended area decrease substantially as desired.

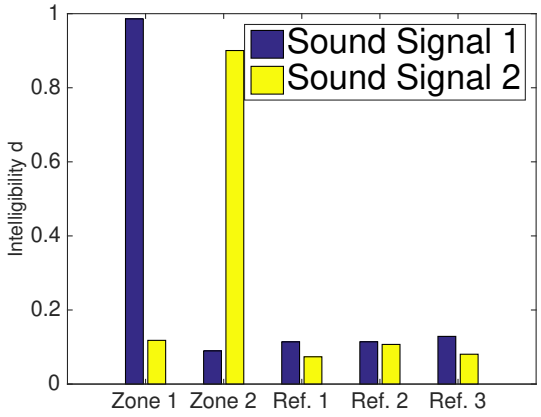

Figure 2.13: STOI intelligibility scores for the final simulation with chopped white noise signals as inputs

2.6 No Chopping: The Least-Squares and the Null Space Approaches

We now present two improved strategies which do not involve any chopping [16]. The reason is that based on the discussions in previous sections and further simulation attempts, we conclude that chopping is not needed. In fact, having white noise as the input for the two approaches is the key.

Null Space (NS) Approach

In the NS approach, the idea is to send random noise from loudspeakers in addition to message signals. The noise signals disappear only at the focusing points while they continue to mask the messages everywhere else. This results in the reception of clean audio messages at the focusing points while having low intelligibility in unintended area. This technique is inspired by standard methods in wireless networking on jamming eavesdroppers [17, 18].

In this approach, the design signal $\boldsymbol{x_k}$ is chosen as a sum of a message-carrying vector $\boldsymbol{s_k}$ and a noise-like signal $\boldsymbol{w_k}$. That is, $\boldsymbol{x_k} = \boldsymbol{s_k} + \boldsymbol{w_k}$. It satifies

$$\boldsymbol{y}_k = \boldsymbol{H}(\boldsymbol{s_k} + \boldsymbol{w_k}) = \boldsymbol{H}\boldsymbol{s_k}, \tag{2.25}$$

where $\boldsymbol{H}$ is a matrix with Toeplitz matrix blocks that contain the overall impulse responses of all the channels. Thus, it is also called the channel matrix.

The task then is to construct $\boldsymbol{s_k}$ and $\boldsymbol{w_k}$ to satisfy $\boldsymbol{H}\boldsymbol{s_k} = \boldsymbol{y_k}$ and $\boldsymbol{H}\boldsymbol{w_k} = \boldsymbol{0}$. This is achieved by choosing $\boldsymbol{w_k}$ as the projection of a random noise vector on the nullspace of the channel matrix $\boldsymbol{H}$, i.e., $\boldsymbol{w_k} = \boldsymbol{P}_{\boldsymbol{\mathcal{N}(H)}}\boldsymbol{v}$, where the entries of $\boldsymbol{v}$ are independent and identically distributed (iid) standard Gaussian and $\boldsymbol{P}_{\boldsymbol{\mathcal{N}(H)}}$ is the projector on the null space of $\boldsymbol{H}$.

As mentioned in the previous section, $\boldsymbol{H}$ is typically of large dimension, which makes the direct computation of its nullspace a prohibitively complex task. Instead, we first find the projection of $\boldsymbol{v}$ on the row space of $\boldsymbol{H}$ by solving

$$\hat{\mathbf{z}} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \ \|\boldsymbol{v} - \boldsymbol{H}^\top \boldsymbol{z}\|_2^2. \tag{2.26}$$

We again use the conjugate gradient method to solve Eq. (2.26) using fast Fourier transforms since $\boldsymbol{H}$ is block-Toeplitz. Once $\hat{\mathbf{z}}$ is found, the nullspace projection $\boldsymbol{P}_{\boldsymbol{\mathcal{N}(H)}}\boldsymbol{v}$ is simply $\boldsymbol{v} - \boldsymbol{H}^\top \hat{\boldsymbol{z}}$.

Comparison: Simulations

In simulations, we randomly place two focusing points inside a $7m \times 8m$ shoebox room and calculate STOI values of the signals arriving at the focusing spots using both the LS method and the NS method. An additional location is randomly chosen to check how degraded the audio signals appear outside the target focusing spots.

This is repeated for two settings: anechoic and reverberant. As we can see in Fig. 2.14, the two approaches both perform well, acheiving private audio delivery. Furthermore, the experiment verifies the importance of the presence of echoes for both approaches.

To see this, Fig. 2.14a shows that under the anechoic setting, the received signal at the first focusing point has nearly perfect STOI values for both approaches. However, the received signal at the second focusing point has reduced intelligibility. On the other hand, Fig. 2.14b shows that in the presence of echoes, signal intelligibility is restored at the second focusing point as well. This indicates that the spatial diversity provided by echoes helps in conditioning the system matrix $\mathbf{H}\tilde{\mathbf{X}}$ for the LS method (or the channel matrix $\boldsymbol{H}$ for the NS method), which in turn supports perfect reconstruction of messages at target locations.

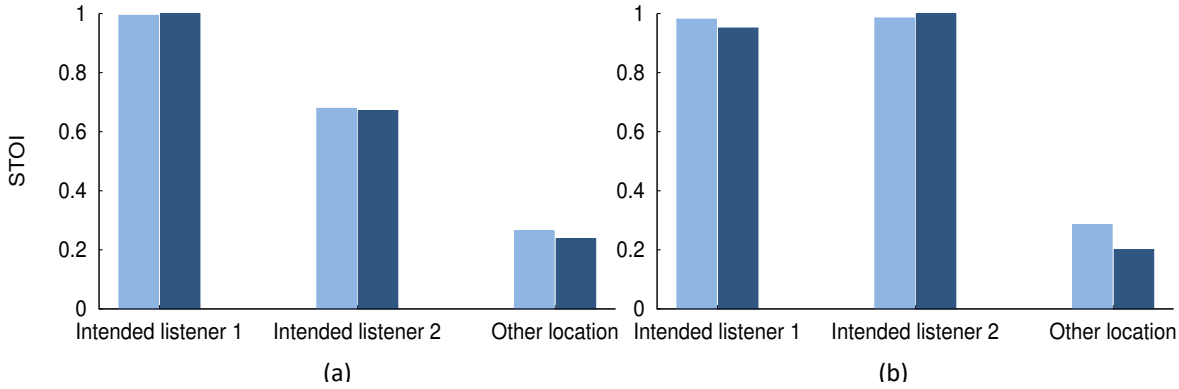

Figure 2.14: Comparison of the two approaches (the LS method in dark blue and the NS method in light blue) under two settings (a) Anechoic (b) Reverberant

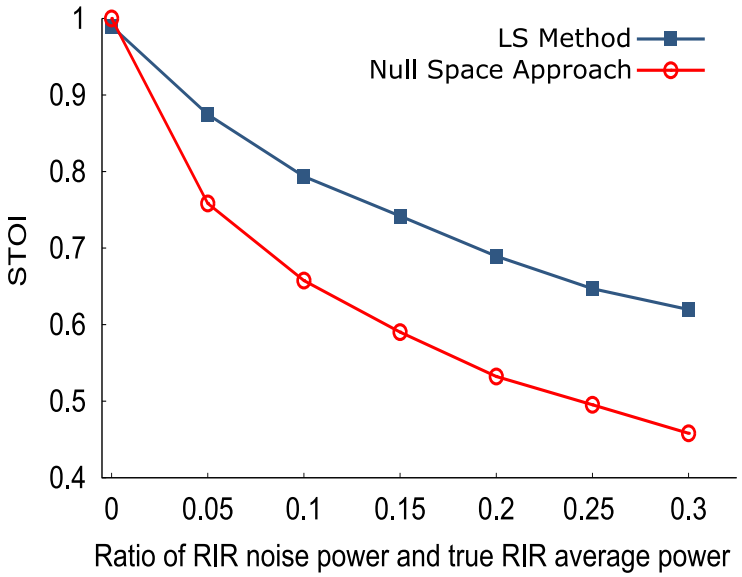

Figure 2.15: Comparison of the two approaches at one focusing point with different levels of noise in the overall impulse response

We further assess how robust the two approaches are against errors in the impulse responses. This is an important metric since the impulse responses will be measured with some degree of uncertainties when we apply the techniques in real world.

In Fig. 2.15, we inject different levels of noise into the impulse responses and measure the corresponding STOI score at one focusing point. As it can be observed, the LS method retains higher STOI score compared with the null space method at the same level of noise injection.

Comparison: Real-World Experiments

In addition to simulations, we also perform an experiment to evaluate the two approaches in a real room with six loudspeakers. We measure the STOI scores of generated sounds at seven locations with microphones. The experimental setup is shown in Fig. 2.16. Two microphones are chosen to be the focusing points, and the

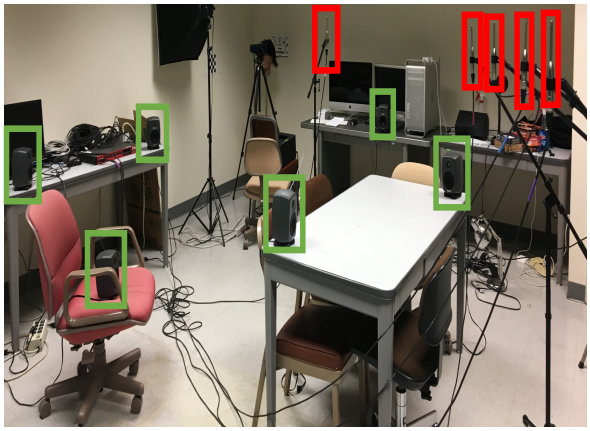

Figure 2.16: Settings for the real-world experiments; red boxes mark the location of the microphones while green boxes mark the location of the loudspeakers

rest are placed at increasing distances from the second focusing point. For the overall impulse response pairs between the loudspeakers and the focusing points, they are measured using the approach described in Farina's paper [19].

Figure 2.17 shows the measured STOI values at these locations. The observed intelligibility at the two spots is good with high STOI values, and the scores considerably degraded 50 cm away from the focusing points. The two approaches again display similar performance while the null space approach has slightly stronger rate of intelligibility decay with respect to distance.

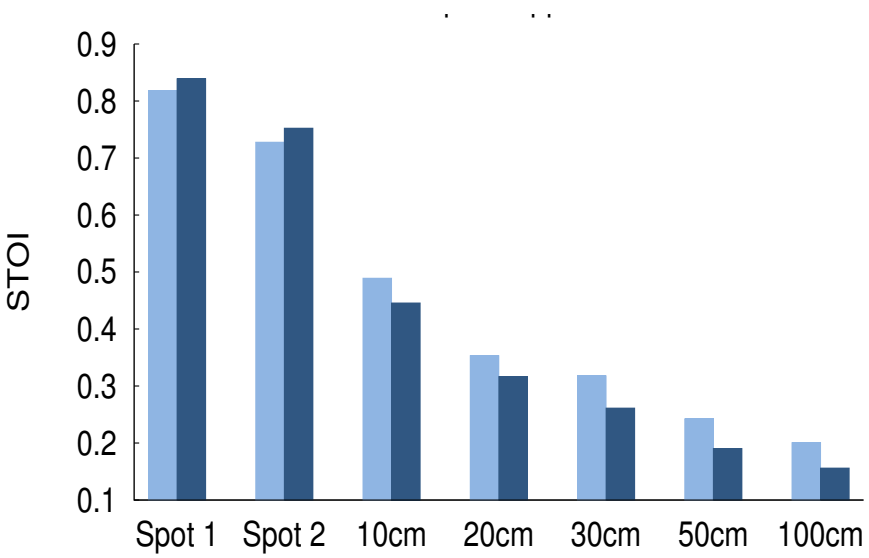

Figure 2.17: Comparison of the two approaches on how intelligibility degrades with respect to distance (the LS method in light blue and the NS method in dark blue)

CHAPTER 3

CONCLUSION

In this thesis, we studied new approaches to the sound zone problem with the added privacy constraint. The new design goal is that no one besides the intended listener should be able to comprehend the message.

The main feature of standard methods which makes it hard to achieve privacy is that all loudspeakers are emitting linearly filtered version of the original signal. This means that even without special equipment one has a decent shot at understanding the individual audio streams, even though they are attenuated outside of the target zone.

This motivated us to look for solutions that avoid emitting filtered versions of the verbatim messages intended for the users. The initial proposal with chopped target signals was based on the intuition that recombining the chopped segments via a combination of designed filters and reverberation will enable a recombination of the pieces at the right places, while precluding an intelligible recombination elsewhere. Through real and numerical experiments it became clear that in the presence of complex multipath, the key ingredient that made this strategy work was the fact that it is noiselike. Hence, in subsequent designes, we turned to using pure noise as the source signals, and derived new approaches that borrow from wireless communications in addition to the previously proposed least-squares designs.

Through a number of numerical and real experiments we showed that the proposed methods indeed achieve the goal of sound focusing at multiple spots, while generating largely unintelligible signals everywhere else. While our algorihtms are

not replacements for the classical sound zone approaches, we believe they pave a promising new way for private audio.

As far as future work goes, we envision influences going in both directions: noise-based methods may help achieve some level of privacy in classical sound-zone methods, as well as other goals such as perceptual masking. Insights from classical sound zone theory could help produce spatially extended focusing zones in noise-based methods. Additionally, the folowing drawbacks should be resolved:

- With high sampling rates, the conjugate-gradient-based design, even with FFT-based optimizations, is still too slow for real-time filter updates.
- Manual impulse response measurements at the focusing spots are impractical and preclude application to unseen spaces.

Notwithstanding, the fact that the methods work as well as they do in real rooms and real experiments gives us reasons for optimism. We believe future research will help make the methods presented in this thesis practical and applicable in a variety of real-world scenarios.

APPENDIX A

SOLVING A SYSTEM OF EQUATIONS

Solving a system of linear equations such as

$$\begin{cases} 2x_1 - 8x_2 + 10x_3 = 10 \\ -9x_1 + 3x_2 + x_3 = -1 \\ 4x_1 - 3x_2 + 2x_3 = 27 \end{cases}$$

arises in numerous situations such as designing a finite impulse response (FIR) digital filter, solving state-space equations of a dynamic control system or simply fitting a line to a set of data.

Having three equations with three unknowns, we know a unique solution exists as long as the left-hand side (LHS) are linearly independent. For example,

$$2x_1 - 8x_2 + 10x_3$$

can be easily verified as a linear combination of

$$4x_1 - 12x_2 + 8x_3 \quad \text{and} \quad -x_1 + 2x_2 + x_3.$$

Also, we know that for any linear equations, a matrix represenation exists. With the example system of linear equations above, we can instead write it in the

matrix form as follows

$$\begin{bmatrix} 2 & 8 & 10 \\ -9 & 3 & 1 \\ 4 & -3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ -1 \\ 27 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{A}$ is called the system matrix.

For this particular case, it can be verified that the square system matrix $\boldsymbol{A}$ has rank 3 due to the linear independence among the equations and is thus invertible. Therefore, we can obtain a unique solution $\hat{\boldsymbol{x}}$ by multiplying both sides with the inverse matrix of $\boldsymbol{A}$, $\boldsymbol{A}^{-1}$

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b} = \begin{bmatrix} -1.33 \\ -6.48 \\ 6.45 \end{bmatrix}.$$

A.1 Least-Squares Problems

What if the system matrix $\boldsymbol{A}$ is not square? First, we want to examine the case when $\boldsymbol{A}$ is a tall matrix (more rows than columns) with independent columns, which means no column is a linear combination of the remaining ones.

Suppose that $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, where $m > n$, and we have the same system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. For this particular system of equations, we have more equations than variables. Therefore, we call this an overdetermined system of equations. With an overdetermined system of equations, it is not guaranteed that an exact solution $\hat{\boldsymbol{x}}$ exists. However, with the given assumption that $\boldsymbol{A}$ has independent columns, if there exists an exact solution $\hat{\boldsymbol{x}}$, it is guaranteed to be unique.

If no exact solution exists, a reasonable idea is to find an approximate solution $\hat{\boldsymbol{x}}$ that gives $\boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}$ which is in a sense "closest" to the $\boldsymbol{b}$ given by the problem. To measure the "closeness" between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, the 2-norm, $\|\cdot\|$, is a common choice. For a vecotr $\mathbf{v}$, the 2-norm of it is defined as

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}, \tag{A.1}$$

where $\langle \cdot, \cdot \rangle$ represents an inner product. Therefore, a mathematical measurement of how close the approximation $\hat{\boldsymbol{b}}$ is to the $\boldsymbol{b}$ can be written as

$$\|\hat{\boldsymbol{b}} - \boldsymbol{b}\| = \|\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b}\|. \tag{A.2}$$

Using this measure of closeness, for the case when no exact solution exists, finding the best approximate solution $\hat{\boldsymbol{x}}$ that minimizes the 2-norm distance between $\hat{\boldsymbol{b}}$ and $\boldsymbol{b}$ is written as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|. \tag{A.3}$$

For mathematical convenience, it is standard to rewrite the problem in the equivalent form below by simply adding a square term to the 2-norm

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2. \tag{A.4}$$

Problems of this type as in Eq. (A.4) are known as least-squares problems.

A.2 Finding the Solution $\hat{\boldsymbol{x}}$

We now explain how to find the least-squares solution to an overdetermined system of linear equations.

Subspace

First, we need the notion of subspace to explain how the solution, $\hat{\boldsymbol{x}}$, is calculated. Suppose a set S contains n vectors $\boldsymbol{a_1}$, $\boldsymbol{a_2}$, ... $\boldsymbol{a_n}$. The subspace $\mathbb{S}$ formed by

the set of vectors is defined as

$$\mathbb{S} = \{s \mid s = c_1\boldsymbol{a_1} + c_2\boldsymbol{a_2} + \cdots + c_n\boldsymbol{a_n} \text{ and } c_1, c_2, \ldots c_n \in \mathbb{C}\}. \tag{A.5}$$

In other words, the subspace $\mathbb{S}$ is the space formed by every possible linear combination of the set of vectors, which is often referred to as the span of $\{\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots \boldsymbol{a_n}\}$.

Back to solving the least-squares problem, given matrix $\boldsymbol{A}$ which has the form

$$\boldsymbol{A} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{a_1} & \boldsymbol{a_2} & \cdots & \boldsymbol{a_n} \\ | & | & \cdots & | \end{bmatrix},$$

the matrix multiplication $\boldsymbol{Ax}$ can be seen exactly as some linear combination of the vectors $\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots \boldsymbol{a_n}$,

$$\boldsymbol{Ax} = x_1\boldsymbol{a_1} + x_2\boldsymbol{a_2} + \ldots + x_n\boldsymbol{a_n}.$$

Therefore, the resulting vector $\boldsymbol{b}$ is said to live in the subspace $\mathbb{S}$ formed by $\{\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots \boldsymbol{a_n}\}$. However, having $\boldsymbol{b}$ live in the subspace $\mathbb{S}$ implies the existence of an exact solution. For a least-squares problem, the given $\boldsymbol{b}$ is often not in the subspace $\mathbb{S}$ while the approximate $\hat{\boldsymbol{b}}$ is always in the subspace $\mathbb{S}$.

With the above constraint for $\hat{\boldsymbol{b}}$, the least-squares problem is converted into finding a point in the subspace $\mathbb{S}$ that is closest to $\boldsymbol{b}$ in the 2-norm sense. The good news is the problem now has a well-known geometric interpretation and the corresponding solution.

Orthogonal Projection

For illustration purpose, suppose we have $\boldsymbol{b} = [x_b, y_b, z_b]^T$ and a subspace $\mathbb{S}$ which is a plane spanned by two independent vectors, the geometry will be similar to what is depicted in Fig. A.1. Here, $\boldsymbol{b}$ is not in $\mathbb{S}$. Therefore, an approximation $\hat{\boldsymbol{b}}$ which resides in $\mathbb{S}$ needs to be found. Also shown in Fig. A.1, the well-known solution to finding a closest point $\hat{\boldsymbol{b}}$ on a plane to another point $\boldsymbol{b}$ is calculating the orthogonal projection of point $\boldsymbol{b}$ onto the plane.

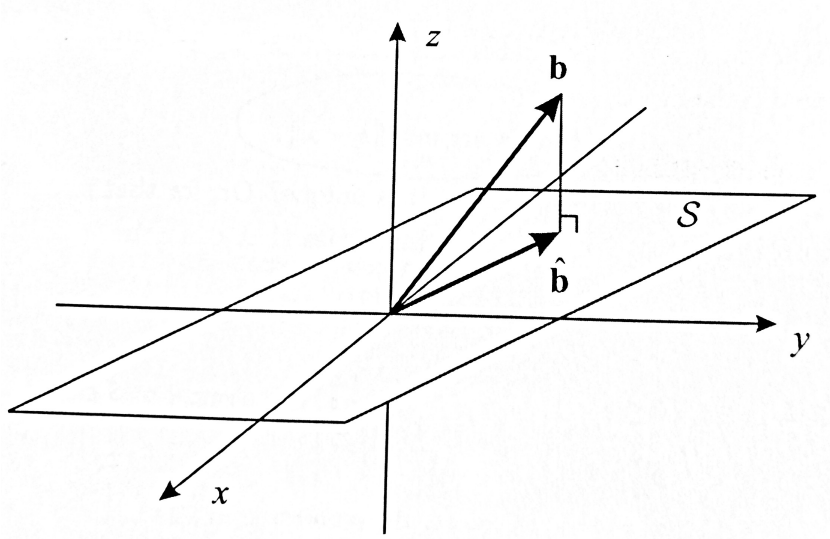

Figure A.1: Projection in three-dimensional space [20]

In the 3-D case shown in Fig. A.1, it is clear that the orthogonal projection $\hat{\boldsymbol{b}}$ is the closest point in $\mathbb{S}$ to $\boldsymbol{b}$. The same orthogonality is valid in arbitrary dimension based on the Orthogonality Principle [20]. The theorem states that $\|\hat{\boldsymbol{b}} - \boldsymbol{b}\| \leq \|\boldsymbol{y} - \boldsymbol{b}\|$ for any $\boldsymbol{y} \in \mathbb{S}$. As a result, the coefficient vector of the orthogonal projection is the unique solution to the least-squares problem.

Suppose we define the orthogonal projector $\boldsymbol{P}_{\mathbb{S}}$ which projects $\boldsymbol{b}$ onto $\mathbb{S}$, which gives us $\boldsymbol{P}_{\mathbb{S}}\boldsymbol{b} = \hat{\boldsymbol{b}}$, then the projector must meet the two properties:

- Being idempotent: $\boldsymbol{P}_{\mathbb{S}}{}^2 = \boldsymbol{P}_{\mathbb{S}}$,

- Being Hermitian: $\boldsymbol{P}_{\mathbb{S}}{}^{*} = \boldsymbol{P}_{\mathbb{S}}$.

Following the above properties, the projector $\boldsymbol{P}_{\mathbb{S}}$ for subspace $\mathbb{S}$ can be shown to be

$$\boldsymbol{P}_{\mathbb{S}} = \boldsymbol{A}(\boldsymbol{A}^{*}\boldsymbol{A})^{-1}\boldsymbol{A}^{*}. \tag{A.6}$$

With the formula for the projector $\boldsymbol{P}_{\mathbb{S}}$, it is now straightforward to find the solution

$$\begin{aligned} \boldsymbol{A}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}} = \boldsymbol{P}_{\mathbb{S}}\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{A}^{*}\boldsymbol{A})^{-1}\boldsymbol{A}^{*}\boldsymbol{b}, \\ \hat{\boldsymbol{x}} = (\boldsymbol{A}^{*}\boldsymbol{A})^{-1}\boldsymbol{A}^{*}\boldsymbol{b}. \end{aligned} \tag{A.7}$$

LS as an Optimization Problem

Instead of using the geometric interpretation and the orthogonal projection method, the same result can also be obtained by considering the least-squares problem in Eq. (A.4) as an optimization problem.

To minimize the convex cost function $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$, it is enough to solve $\nabla_{\boldsymbol{x}}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = 0$ to obtain the global minimum.

Expanding the terms,

$$\begin{aligned} \nabla_{\boldsymbol{x}}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 &= \nabla_{\boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^{*}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \nabla_{\boldsymbol{x}}(\boldsymbol{x}^{*}\boldsymbol{A}^{*}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^{*}\boldsymbol{b} - \boldsymbol{x}^{*}\boldsymbol{A}^{*}\boldsymbol{b} - \boldsymbol{b}^{*}\boldsymbol{A}\boldsymbol{x}) \\ \text{(Assuming only real numbers)} &= \nabla_{\boldsymbol{x}}(\boldsymbol{x}^{*}\boldsymbol{A}^{*}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^{*}\boldsymbol{b} - 2\boldsymbol{x}^{*}\boldsymbol{A}^{*}\boldsymbol{b}) \\ &= 2\boldsymbol{A}^{*}\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^{*}\boldsymbol{b} \\ &= 0. \end{aligned}$$

Rearranging the terms gives

$$\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^*\boldsymbol{b}. \tag{A.8}$$

Equation (A.8) is famously known as the normal equation. Solving the normal equation gives the same solution as the orthogonal projection method,

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*\boldsymbol{b}. \tag{A.9}$$

Moore-Penrose Pseudoinverse

Up to this point, we insisted on having a closed-form expression for the solution $\hat{\boldsymbol{x}}$ that minimizes the 2-norm distance $\|\hat{\boldsymbol{b}} - \boldsymbol{b}\|$. In general, for any system of equations (overdetermined, underdetermined or square), an expression exists for an (approximate or exact) solution $\hat{\boldsymbol{x}}$.

Let us define a matrix $\boldsymbol{A}^\dagger$, called the Moore-Penrose pseudoinverse or simply pseudoinverse of $\boldsymbol{A}$, which meets the so-called Penrose conditions:

- $\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{A}$,
- $\boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}^\dagger$,
- $(\boldsymbol{A}\boldsymbol{A}^\dagger)^* = \boldsymbol{A}\boldsymbol{A}^\dagger$,
- $(\boldsymbol{A}^\dagger\boldsymbol{A})^* = \boldsymbol{A}^\dagger\boldsymbol{A}$.

$\boldsymbol{A}^\dagger$ is treated as a generalization of an inverse matrix for any matrix, square or non-square. More specifically, to solve for the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, a concise expression is given as

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^\dagger\boldsymbol{b}. \tag{A.10}$$

In general, singular value decomposition (SVD) is needed to compute the pseudoinverse of an arbitrary matrix $\boldsymbol{A}$. Closed-form expressions for $\boldsymbol{A}^\dagger$ can be derived if matrix $\boldsymbol{A}$ is full-rank.

For underdetermined systems of equations with a full-ranked fat matrix $\boldsymbol{A}$, the least-norm solution $\hat{\boldsymbol{x}}$ can be obtained by using

$$\boldsymbol{A}^\dagger = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}. \tag{A.11}$$

On the other hand, for overdetermined systems of equations with full-ranked tall matrix $\boldsymbol{A}$, the pseudoinverse is derived as

$$\boldsymbol{A}^\dagger = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*. \tag{A.12}$$

This result again leads to the least-squares solution shown in Eq. (A.9).

A.3 Direct Solvers

For a system matrix $\boldsymbol{A}$ of a small size, it is computationally inexpensive to compute the solution $\hat{\boldsymbol{x}}$. For large systems, however, computing the inverse of $\boldsymbol{A}$ directly is either expensive or impossible. Therefore, efficient direct solvers are more often utilized.

LU Decomposition

The first technique to avoid the direct computation of an inverse is the lower-upper triangular (LU) decomposition. The method factorizes a square matrix into a product of a lower triangular matrix $\boldsymbol{L}$ and a upper triangular matrix $\boldsymbol{U}$. Using a three by three matrix $\boldsymbol{A}$ as an example,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & & 0 \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ & U_{22} & U_{23} \\ 0 & & U_{33} \end{bmatrix} = \boldsymbol{LU}. \tag{A.13}$$

Solving a system of equation $\boldsymbol{Ax} = \boldsymbol{b}$ then becomes

$$\begin{bmatrix} L_{11} & & 0 \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ & U_{22} & U_{23} \\ 0 & & U_{33} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} L_{11} & & 0 \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \boldsymbol{y} = \boldsymbol{b}, \tag{A.14}$$

where $\boldsymbol{y} = \boldsymbol{Ux}$.

It can be observed that solving for $\boldsymbol{y}$ is a trivial task since $\boldsymbol{L}$ is a lower triangular matrix. The solution can be readily found using a top-to-bottom approach. Once $\boldsymbol{y}$ is found, the goal is then changed to solving the equation $\boldsymbol{Ux} = \boldsymbol{y}$. Again, with matrix $\boldsymbol{U}$ as an upper triangular matrix, finding the solution $\boldsymbol{x}$ is simple. It can be found using a bottom-to-top approach.

How is the LU decomposition relevant to solving least-squares problems where the system matrix $\boldsymbol{A}$ is not even square? Recall that solving a least-squares problem equals solving for the solution of the normal equation

$$\boldsymbol{A}^*\boldsymbol{Ax} = \boldsymbol{A}^*\boldsymbol{b}. \tag{A.15}$$

The product $\boldsymbol{A}^*\boldsymbol{A}$ is a square matrix. Therefore, solving the equation is essentially solving a system of equations with a square system matrix in the form

$$\hat{\boldsymbol{A}}\boldsymbol{x} = \hat{\boldsymbol{b}}, \tag{A.16}$$

where the square matrix $\hat{\boldsymbol{A}} = \boldsymbol{A}^*\boldsymbol{A}$ and $\hat{\boldsymbol{b}} = \boldsymbol{A}^*\boldsymbol{b}$. Therefore, the LU decomposition method can be applied.

Algorithms exist for efficiently finding the LU decomposition of a square matrix. As a result, it can be a powerful direct solver for a least-squares problem without directly solving for the inverse of $\boldsymbol{A}^*\boldsymbol{A}$.

Cholesky Decomposition

Similar to the idea of the LU decomposition, the Cholesky decomposition breaks a matrix down into a product of matrices that have nice properties for finding the solution when the matrix is the system matrix $\boldsymbol{A}$ in $\boldsymbol{Ax} = \boldsymbol{b}$. The Cholesky decomposition may be seen as the LU decomposition with some constraints on the matrix of interest. Namely, the matrix to decompose must be Hermitian and positive definite. The tradeoff of having constraints is compensated by the fact that using the Cholesky decomposition to solve a system of equations can be twice as fast as using the LU decomposition [21].

Instead of having the general decomposition form $\boldsymbol{A} = \boldsymbol{LU}$, the Cholesky decomposition gives us $\boldsymbol{A} = \boldsymbol{LL}^*$. To solve least-squares problems, the same substitution trick introduced in Eq. (A.16) and the same procedures are used.

A.4 Iterative Solvers

Even though having a closed-form solution might suggest that our work is done, in practice, the problem size is often too big to allow for computation of matrix inverses. Direct solvers mentioned in the previous subsection are sufficient to solve a system of equations when the system matrix $\boldsymbol{A}$ is of relatively small size. However, in practice, this is often not the case.

For example, with a high pixel count, many operations on an image can be represented by matrix operations on huge matrices. When processing audio, matrices containing time domain signals can grow very large at sampling rates of 44.1 kHz or 48 kHz. These situations then warrant the use of iterative methods which, instead

of computing an inverse, directly compute $\boldsymbol{A}$.

Gradient Descent Method

In Eq. (A.9), it is shown that considering least-squares problems as optimization problems gives us the solution $\hat{\boldsymbol{x}}$ that equals the one obtained from using the orthogonal projection method. Therefore, iterative solvers used for optimization problems naturally become the tools of choice for solving least-squares problems.

One of the methods most commonly used is the gradient descent method. Even though the method does not exhibit a particularly favorable convergence rate, it can be very easily implemented. In the recursive form, the method can be seen as computing

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - \alpha \nabla f(\boldsymbol{x_k}), \tag{A.17}$$

with a suitable choice of the starting point $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$ and where $f(\boldsymbol{x_k})$ is an objective function of $\boldsymbol{x_k}$.

Here, α, which is termed the learning rate, can be chosen based on a line search at the start of each iteration or fixed throughout. The largest allowable α is determined by the spectrum of $\boldsymbol{A}$.

To obtain an intuitive idea behind the gradient descent method, a simple one-dimensional case is shown in Fig. A.2.

In Fig. A.2, a function $f(x)$ is given by the black curve. Assuming this function $f(x)$ is convex, which means the function $f(x)$ satisfies the condition

$$\begin{aligned} (1-\alpha)f(x_1) + \alpha f(x_2) \geq f((1-\alpha)x_1 + \alpha x_2), \\ \text{where} \quad 0 \leq \alpha \leq 1, \end{aligned} \tag{A.18}$$

it is then guaranteed that all local minima are global minima. Suppose that the $(x_1, f(x_1))$ pair was given, how can the optimal pair $(x^*, f(x^*))$ be found itera-

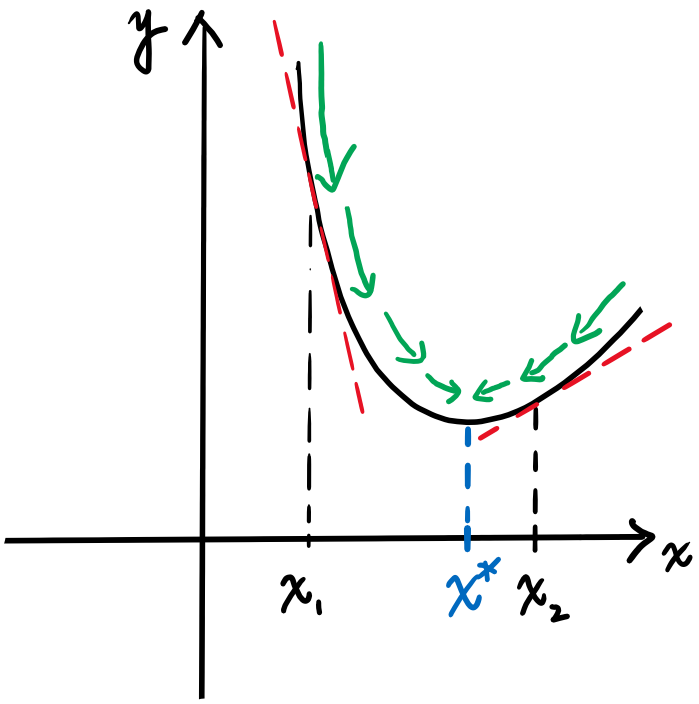

Figure A.2: Gradient descent for a two-dimensional case

tively?

First, it can be observed that the gradient, or simply the derivative in this one-dimensional case, will be negative whenever x_1 is smaller than x^* and will be positive whenever x_1 is greater than x^*. Therefore, it makes sense to add the term $-\frac{df(x_1)}{dx}$ or rather more generally the term $-\nabla f(\boldsymbol{x_1})$ to x_1 for computing the next term x_2. With a scaling factor α, the simple but effective recursive search becomes the gradient descent formula shown in Eq. (A.17). With proper α chosen, the method will eventually converge to x^*.

In Fig. A.3, the iterative steps of the gradient descent method are shown for the two-dimensional case with a starting point at $\boldsymbol{x}_{(0)}$ together with the line search strategy for choosing α. In this case, the learning rate α which determines the length

of each search step is computed at the start of each iteration using a line search. To compensate for the extra computation efforts on line search, the benefit gained is the reduced convergence time. For this particular case, exactly six iterations are needed to converge to the minimum.

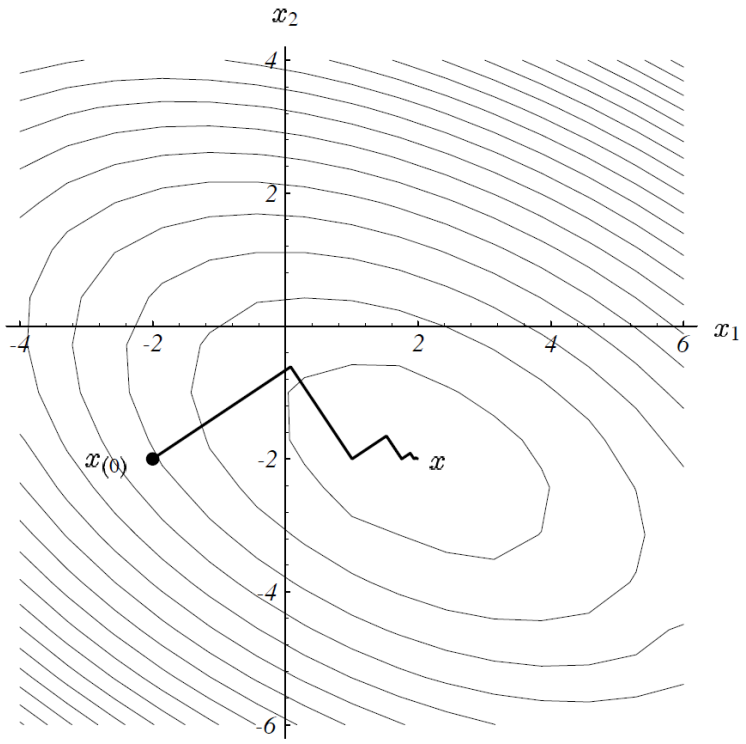

Figure A.3: Gradient descent method using line search on α to find the minimum of a quadratic function in 2-D [22]

The idea behind line search is that at the k^{th} step, we look for $f(\boldsymbol{x_{k+1}})$ that minimizes $\|f(\boldsymbol{x_{k+1}})\|$. By doing this, the objective is to achieve a better convergence rate. In other words, the line search problem can be stated as

$$\alpha_{best} = \arg\min_{\alpha} \|f(\boldsymbol{x_k} - \alpha \nabla f(\boldsymbol{x_k}))\|. \tag{A.19}$$

To minimize the expression in Eq. (A.19), the gradient at the end point,

$\nabla f(\boldsymbol{x_k} - \alpha \nabla f(\boldsymbol{x_k}))$ needs to be considered. For example, during the first iteration in the case shown in Fig. A.3, the starting point $\boldsymbol{x}_{(0)}$ and its gradient $\nabla f(\boldsymbol{x}_{(0)})$ are given. The search line formed by $\nabla f(\boldsymbol{x}_{(0)})$ can be seen as the gray line in Fig. A.4. The end point $\boldsymbol{x}_{(1)}$ would then be a point on this line. The solid arrows in Fig. A.4 represent the gradient at those specific points. Along with those gradients are the projections of the gradients onto the search line. It then can be observed that at the point $\boldsymbol{x}_{(1)}$ which minimizes $f(\boldsymbol{x}_{(1)})$, the gradient $\nabla f(\boldsymbol{x}_{(1)})$ is orthogonal to the search direction $\nabla f(\boldsymbol{x}_{(0)})$. Therefore, in general for the k^{th} step, the line search becomes a task of finding α that satisfies

$$\nabla f(\boldsymbol{x}_k) \cdot \nabla f(\boldsymbol{x}_{k+1}) = \nabla f(\boldsymbol{x}_k) \cdot \nabla f(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)) = 0. \tag{A.20}$$

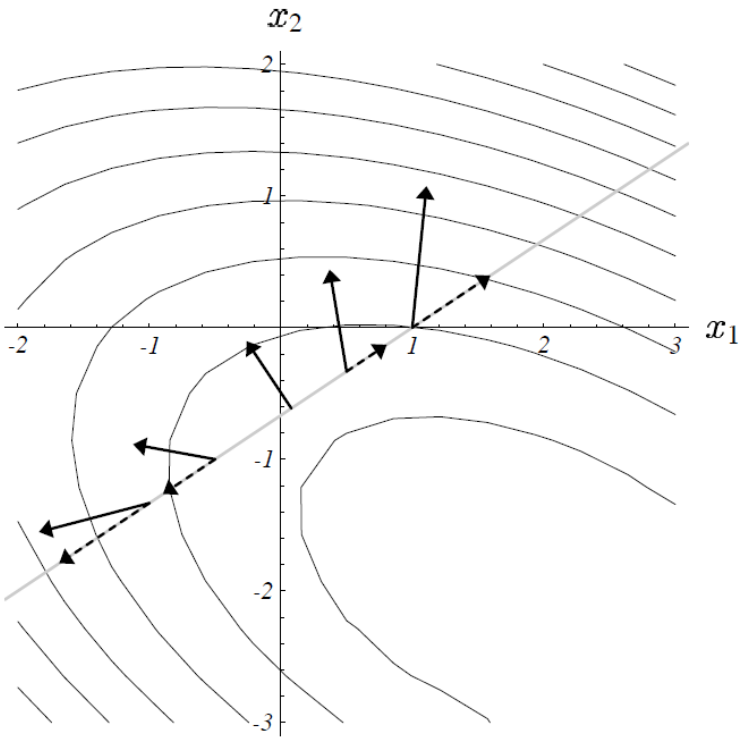

Figure A.4: How line search works on finding the best α [22]

Depending on the complexity of the gradient computation, it might not be

deemed worthwhile to utilize line search at each iteration. Still, line search is commonly seen in optimization problems as it reduces the convergence time when compared with less principled ways of choosing α.

Although the gradient descent method is guaranteed to converge given a convex function $f(\boldsymbol{x})$, the convergence is often slow even with line search implemented. This fact can be seen by the zig-zag pattern of the search in Fig. A.3. The reason behind it is that each iteration of gradient descent does not "do its best" in terms of spanning the space.

For example, in Fig. A.3 where the function is in two-dimensional space, the odd number searches actually are all in the same direction, whereas the even number searches are also all in the same direction. It would appear that combining the searches of the same direction into one iteration is more efficient and less redundant (redundancy of the search causes the zig-zag). Therefore, in an example where each iteration "does its best", the two-dimensional space can be spanned by merely two searches, and the minimum can be found with fewer iterations. This is the main improvement conjugate gradient method provides over the gradient descent method.

Conjugate Gradient Method

Following the above discussion, how do we decide on the search direction and the length of each search to maximize the effect of each iteration? The first naive attempt might be trying to make each search direction orthogonal to each other. In other words, $\boldsymbol{d_i}^T \boldsymbol{d_j} = 0$ where $i \neq j$ and $\boldsymbol{d_i}$ is the search direction of the i^{th} iteration. This indicates that $\boldsymbol{d_k}^T \boldsymbol{e_{k+1}} = 0$ where $\boldsymbol{e_{k+1}}$ is the error $\boldsymbol{x^*} - \boldsymbol{x_{k+1}}$ after the k^{th} iteration. Following the update rule $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - \alpha_k \boldsymbol{d_k}$, the errors have the relationship

$$\boldsymbol{e_{k+1}} = \boldsymbol{e_k} + \alpha_k \boldsymbol{d_k}. \tag{A.21}$$

However, computing α_k by expanding the terms results in

$$\begin{aligned} \boldsymbol{d_k}^T \boldsymbol{e_{k+1}} &= \boldsymbol{d_k}^T(\boldsymbol{e_k} + \alpha_k \boldsymbol{d_k}) \\ &= \boldsymbol{d_k}^T \boldsymbol{e_k} + \alpha_k \boldsymbol{d_k}^T \boldsymbol{d_k} = 0, \\ \alpha_k &= -\frac{\boldsymbol{d_k}^T \boldsymbol{e_k}}{\boldsymbol{d_k}^T \boldsymbol{d_k}}. \end{aligned} \tag{A.22}$$

It turns out that α_k depends on the error $\boldsymbol{e_k}$. However, $\boldsymbol{e_k}$ has to be computed based on $\boldsymbol{x^*}$, which is the quantity we are aiming to compute. Instead, a different notion of orthogonality is used, called A-orthogonality.

Before A-orthogonality is discussed, the term Krylov subspace should be defined first. An i^{th} dimension Krylov subspace $\boldsymbol{K_i}$ is defined as

$$\boldsymbol{K_i} = span[\boldsymbol{v}, \boldsymbol{W}\boldsymbol{v}, \cdots, \boldsymbol{W^{i-1}}\boldsymbol{v}], \tag{A.23}$$

where $\boldsymbol{W}$ is any matrix, and $\boldsymbol{v}$ is any vector. Therefore, the subspace $\boldsymbol{K_i}$ very much depends on the choice of $\boldsymbol{W}$ and $\boldsymbol{v}$. The conjugate gradient method is essentially an iterative method based on solving the expanding Krylov subspace [23]. The Krylov subspace, in particular for solving the system of equations $\boldsymbol{Ax} = \boldsymbol{b}$, has the form

$$\boldsymbol{K_i} = span[\boldsymbol{b}, \boldsymbol{Ab}, \cdots, \boldsymbol{A^{i-1}}\boldsymbol{b}]. \tag{A.24}$$

The j-th iteration of the conjugate gradient method approximates the solution $\boldsymbol{x^*}$ by finding the so-called Krylov sequence as

$$\boldsymbol{x_j} = \arg\min_{\boldsymbol{x} \in \boldsymbol{K_j}} \|\boldsymbol{x} - \boldsymbol{x^*}\|_{\boldsymbol{A}}^2, \tag{A.25}$$

where $\|\cdot\|_{\boldsymbol{A}}$ is the A-norm. Two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are said to be A-orthogonal (or conjugate) if they satisfy

$$\boldsymbol{u}^T \boldsymbol{A}\boldsymbol{v} = 0. \tag{A.26}$$

Therefore, the Krylov sequence in Eq. (A.25) can also be rewritten as

$$\boldsymbol{x_j} = \arg\min_{\boldsymbol{x}\in\boldsymbol{K_j}} (\boldsymbol{x} - \boldsymbol{x}^*)^T \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*). \tag{A.27}$$

We see that for each iteration, the Krylov subspace is expanded by one dimension, and $\boldsymbol{x_j}$ is computed to approximate the solution to $\boldsymbol{Ax} = \boldsymbol{b}$. Because $\boldsymbol{x_j}$ is in the corresponding Krylov subspace $\boldsymbol{K_j}$, the conjugate gradient method forces a constraint on the residual $\boldsymbol{r_j} = \boldsymbol{b} - \boldsymbol{Ax_j}$ that $\boldsymbol{r_j}$ should be orthogonal to all vectors in $\boldsymbol{K_j}$. This way, it theoretically means the algorithm converges after exactly n steps, where n is the dimension of the solution $\boldsymbol{x}^*$.

Following the constraint that $\boldsymbol{r_j}$ should be orthogonal to all vectors in $\boldsymbol{K_j}$ and also the fact that Krylov subspaces obey $\boldsymbol{K_j} \supset \boldsymbol{K_i}$ for $i < j$, it means

$$\boldsymbol{r_i}^T \boldsymbol{r_j} = 0, \qquad i \neq j. \tag{A.28}$$

Further, in a similar fashion, $(\boldsymbol{r_j} - \boldsymbol{r_{j-1}})$ is then orthogonal to $\boldsymbol{K_i}$ for any $i < j$. Because each Krylov sequence $\boldsymbol{x_j}$ is chosen from $\boldsymbol{K_j}$, $(\boldsymbol{x_j} - \boldsymbol{x_{j-1}}) \in \boldsymbol{K_j}$ is true. Therefore, combining the two facts, the relationship can be derived for the two differences:

$$(\boldsymbol{x_i} - \boldsymbol{x_{i-1}})^T (\boldsymbol{r_j} - \boldsymbol{r_{j-1}}) = 0, \qquad i < j. \tag{A.29}$$

Substituting in the relationship $(\boldsymbol{r_j} - \boldsymbol{r_{j-1}}) = \boldsymbol{A}(\boldsymbol{x_j} - \boldsymbol{x_{j-1}})$, Eq. (A.29) becomes the key condition in the conjugate gradient method:

$$(\boldsymbol{x_i} - \boldsymbol{x_{i-1}})^T \boldsymbol{A}(\boldsymbol{x_j} - \boldsymbol{x_{j-1}}) = 0, \qquad i < j. \tag{A.30}$$

Equation (A.30) implies that each of the search direction (gradient) is conjugate to

all the previous search directions, hence the name conjugate gradient method.

Using the conditions described in Eq. (A.28) and Eq. (A.30) and proper initialization parameters, the direction vectors $\boldsymbol{d_i}$ and the scaling factors $\boldsymbol{\alpha_i}$ can be iteratively computed with all known values.

One great property of the conjugate gradient method is the convergence rate. It is well known that given a well-conditioned $\boldsymbol{A}$, the conjugate gradient method converges very fast because the number of iterations required is proportional to $\sqrt{\kappa}$, where κ is the condition number of $\boldsymbol{A}$.

REFERENCES

[1] T. Betlehem, W. Zhang, M. A. Poletti, and T. D. Abhayapala, "Personal sound zones: Delivering interface-free audio to multiple listeners," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 81–91, Feb 2015.

[2] J.-H. Chang, C.-H. Lee, J.-Y. Park, and Y.-H. Kim, "A realization of sound focused personal audio system using acoustic contrast control," *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2091–2097, Apr 2009.

[3] J.-W. Choi and Y.-H. Kim, "Generation of an acoustically bright zone with an illuminated region using multiple sources," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1695–1700, Apr 2002.

[4] Y. J. Wu and T. D. Abhayapala, "Theory and design of sounfield reproduction using continuous loudspeaker concept," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 107–116, January 2009.

[5] S. Yon, M. Tanter, and M. Fink, "Sound focusing in rooms: The time-reversal approach," *The Journal of the Acoustical Society of America*, vol. 113, no. 3, pp. 1533–1543, March 2003.

[6] P. Coleman, P. J. B. Jackson, and M. Olik, "Acoustic contrast, planarity and robustness of sound zone methods using a circular loudspeaker array," *The Journal of the Acoustical Society of America*, vol. 135, no. 4, pp. 1929–1940, April 2014.

[7] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *The Journal of the Acoustical Society of America*, vol. 93, no. 5, pp. 2764–2778, May 1993.

[8] W. Jin, W. B. Kleijn, and D. Virette, "Multizone soundfield reproduction using orthogonal basis expansion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 311–315.

[9] D. B. Ward and T. D. Abhayapala, "Reproduction of a plane-wave sound field using an array of loudspeakers," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 6, pp. 697–707, Sep 2001.

[10] W. Jin and W. B. Kleijn, "Theory and design of multizone soundfield reproduction using sparse methods," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2343–2355, Sep 2015.

[11] T. Betlehem and T. D. Abhayapala, "Theory and design of sound field reproduction in reverberant rooms," *The Journal of the Acoustical Society of America*, vol. 117, no. 4, pp. 2100–2111, April 2005.

[12] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 2, pp. 145–152, Feb 1988.

[13] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 4214–4217.

[14] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," 2018.

[15] Y.-J. Liu, J. Casebeer, and I. Dokmanić, "Cocktails, but no party: Multipath-enabled private audio," in *IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018.

[16] A. Chaman, Y.-J. Liu, J. Casebeer, and I. Dokmanić, "Multipath-enabled private audio with noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[17] R. Negi and S. Goel, "Secret communication using artificial noise," in *IEEE Vehicular Technology Conference*, vol. 62, no. 3. Citeseer, 2005, p. 1906.

[18] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, 2008.

[19] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*. Audio Engineering Society, 2000.

[20] Y. Bresler, S. Basu, and C. Couvreur, *Hilbert Spaces and Least Squares Methods for Signal Processing*, Urbana, IL, 2017.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University England EPress, 1992.

[22] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Pittsburgh, PA, USA, Tech. Rep., 1994.

[23] G. Strang, *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.